16th International Conference on Space Operations, Cape Town, South Africa - 03-05 May 2021.

SpaceOps-2021,6,x1347

PINTA – one Tool to plan them all

Rainer Nibler[a]*, Jens Hartung[b], Jonas Krenss[c], Anna Fürbacher[d], Falk Mrowka[e], Sandra Brogl[f]

[a] *Deutsches Zentrum für Luft- und Raumfahrt, GSOC, Weßling, Bayern, 82234, Germany*, Rainer.Nibler@dlr.de
[b] *Deutsches Zentrum für Luft- und Raumfahrt, GSOC, Weßling, Bayern, 82234, Germany*, Jens.Hartung@dlr.de
[c] *Deutsches Zentrum für Luft- und Raumfahrt, GSOC, Weßling, Bayern, 82234, Germany*, Jonas.Krenss@dlr.de
[d] *Deutsches Zentrum für Luft- und Raumfahrt, GSOC, Weßling, Bayern, 82234, Germany*, Anna.Fuerbacher@dlr.de
[e] *Deutsches Zentrum für Luft- und Raumfahrt, GSOC, Weßling, Bayern, 82234, Germany*, Falk.Mrowka@dlr.de
[f] *Spaceopal GmbH, München, Bayern, 80336, Germany*, sandra.brogl@spaceopal.com
* Corresponding Author

Abstract

In the recent years, the "Program for INteractive Timeline Analysis" PINTA, developed at the German Space Operation Center (GSOC), was continuously improved and experienced several evolution steps. PINTA is a GUI application running on Windows-based computer systems, whose main purpose is to serve as the anchor tool for a mission planning operation's engineer when generating, modifying or analysing a mission timeline. This is supported by calling automatic planning algorithms of the embedded generic planning library "PLAnningTOol" PLATO, using input of the embedded orbit propagation and event calculation library "SpaceCraft Orbit and GroundTrack Analysis Tool" SCOTA, or its expandability through plugins.

PINTA is the generic basis of many semi-automated mission planning systems for past, current and future spacecraft projects operated at GSOC. It is used or has been used for the missions Grace, TET-OOV, FireBird, Grace-FollowOn, Eu:CROPIS and is currently prepared for CubeL. Furthermore, PINTA serves as the timeline analysis tool for validating the TerraSAR-X/TanDEM-X mission planning system.

The variety of use cases was further extended to support Launch and Early Orbit Phases (LEOPs) in its special "SoEEditor" configuration as the new generic editing tool for the so-called "Sequence of Events". It was successfully used for the satellites Biros, HAG-1, PAZ, Grace-FollowOn 1 & Grace-FollowOn 2, Eu:Cropis, EDRS-C and is currently in preparation for EnMAP. In addition to LEOP's, the SoEEditor was also capable of supporting the constellation maneuvers for the TerraSAR-X/TanDEM-X mission.

Besides all these use cases, the paper at hand will especially describe how PINTA was even further extended to not only tackle spacecraft-based but also ground-based scheduling. On the one hand it serves as an "On-Call Tool" to support the on-call shifts by automatically generating conflict-free role-based shift plans for all subsystems by considering various constraints like person outages, working hours, role-conflicts, etc… The plan can then be further adapted manually to cope with user change-requests. On the other hand it is used as a "Multi-Mission-Control-Room-and-pass-Scheduler" (MuMiCoRoS) to coordinate the ground-station booking of all LEO (low-earth orbit) satellites: TerraSAR-X, TanDEM-X, TET, Biros, Grace-FollowOn 1 & 2 and Eu:CROPIS. In order to avoid ground-station and operator conflicts between the missions, an automatic and combined plan for all satellites is generated which can then be further modified manually if necessary.

As another use case, PINTA (a.k.a. GPT; Galileo Planning Tool) supports the Galileo Service Operation (GSOp). The planning process involves three timelines: a Short-Term Plan (STP), covering the next ten days, two Mid-Term Plans (MTP) for the Operational (OPE) and the Validation (VAL) chain), covering the next 15 weeks, and a Long-Term Plan (LTP), covering the next 15 months. The activities in these timeframes cover all subsystems of Galileo: Flight Ops, Control segment, Mission segment, remote sites, service operations, hardware, software, hosting, network, etc ... In order to support the GSOp, numerous additional features, like importers, exporters, interfaces and plugins had to be added to PINTA.

Keywords: PINTA, Planning Systems, Sequence of Events, Galileo Planning Tool

Acronyms/Abbreviations

BIROS	-	Bispectral InfraRed Optical System
CCSDS	-	Consultive Committee for Space Data Systems
DLR	-	Deutsches Zentrum für Luft- und Raumfahrt e.V.
EDRS-C	-	European Data Relay Satellite C as part of EDRS (European Data Relay System)
Eu:CROPIS	-	Euglena and Combined Regenerative Organic-food Production in Space
GCC	-	Galileo Control Center
GEO	-	Geostationary Orbit
GfR	-	Gesellschaft für Raumfahrtanwendungen mbH
GPT	-	Galileo Planning Tool
GNSS	-	Global Navigation Satellite System
GRACE	-	Gravity Recovery and Climate Experiment
GFO	-	Gravity Recovery and Climate Experiment – Follow On
GS	-	Ground Station
GSA	-	GNSS Service Agency
GSOC	-	German Space Operations Center
GSOp	-	Galileo Service Operation
GUI	-	Graphical User Interface
HAG-1	-	Hispasat Advanced Generation 1, first Satellite of the OHB SmallGEO generation
LEO	-	Low Earth Orbit
LEOP	-	Launch and Early Orbit Phases
LTP	-	Long-Term Plan
OPE	-	Operational chain
PINTA	-	Program for INteractive Timeline Analysis
PLATO	-	Planning Tool
FOP	-	Flight Operation Procedure (PROTOS Output (Reference))
MMU	-	Mass Memory Unit
MPS	-	Mission Planning System
MTP	-	Mid-Term-Plan
MuMiCoRoS	-	Multi-Mission-Control-Room-and-pass-Scheduler
PAZ-1	-	first Spanish Earth radar observation satellite
PR	-	Planning Request
SCOTA	-	SpaceCraft Orbit and groundTrack Analysis tool
SDT	-	Service desk tool
SoE	-	Sequence of Events
SPOT	-	Swath Preview and Ordering Tool
SSF	-	Saved Stack File (SCOS/GECCOS Output - Reference)
STP	-	Short-Term-Plan
TanDEM-X	-	TerraSAR-X-Add-on for Digital Elevation Measurements, "X" for the X frequency band
TerraSAR-X	-	Terra (Latin for Earth), SAR: Synthetic Aperture Radar, "X" for the X frequency band
TET	-	Technologie-Erprobungsträger
TSTD	-	TerraSAR-X/TanDEM-X
VAL	-	Validation chain

I. Introduction

The application "**PINTA**" has a history of up to two decades serving for various spacecraft missions operated at the "**German Space Operations Center**" (GSOC). It went through a constant change process which was necessary due to new missions with different requirements. In order to get a better overview of how PINTA and the surrounding mission planning framework, that complements its usability, has evolved, the key components will be first described within chapter II below.

Because PINTA was originally intended to serve as the anchor tool for a "**Mission Planning System**" (MPS), which helps the operations engineer when generating, modifying or analysing a mission timeline, chapter III will deal with this in more detail. This will contain all past and current mission planning systems in which PINTA has played or is still playing a major role.

Another area of application of PINTA, which has been established in recent years, is its "**SoEEditor**" functionality for preparing and maintaining the Sequence of Events before and during the launch and early orbit phases (LEOP) for low-earth as well as geostationary satellite missions at GSOC. Its use case as a SoEEditor will be described in more detail in chapter IV.

Although PINTA's original purpose was to plan satellite timelines, the paper at hand will especially describe how it was made possible, to not only tackle spacecraft-based but also ground-based scheduling problems.

On the one hand it serves as an "**On-Call Tool**" to support the on-call shifts by automatically generating conflict-free role-based shift plans for all subsystems by considering various constraints like person outages, working hours, role-conflicts, etc… which will be the topic in chapter V.

On the other hand it is used as a "**Multi-Mission-Control-Room-and-pass-Scheduler**" (MuMiCoRoS) to coordinate the ground-station booking of all LEO (low-earth orbit) satellites: TerraSAR-X, TanDEM-X, TET, Biros, Grace-FollowOn 1 & 2 and Eu:CROPIS. In order to avoid ground-station and operator conflicts between the missions, an automatic and combined plan for all satellites is generated which can then be further modified manually if necessary. See chapter VI for more details.

PINTA, among various competitors, was the tool of choice by the Spaceopal Planning Team, to support the Galileo Service Operation (GSOp) since 2018. This led to a further and previously unknown area of application in which it is now also used as the "**Galileo Planning Tool**". Chapter VII gives a deeper insight into this completely new use case and the associated challenges.

Last but not least, in chapter VIII, an outlook to the future challenges and milestones for the development of PINTA, as well as further projects at GSOC, like the successor "**PintaOnWeb**", is given.

II. Components

2.1. PINTA

The application PINTA is named after the fastest of the three ships used by Christopher Columbus in his first voyage across the Atlantic Ocean, La Pinta ("the spotted one") [12] and is also an abbreviation for "**Program for INteractive Timeline Analysis**".

PINTA is an interactive GUI application running on Windows-based computer systems. Its main purpose is to serve as the anchor tool for a mission planning operation's engineer when generating, modifying, validating or analysing a mission timeline. An example, how a timeline of the TerraSAR-X/TanDEM-X mission can be visualized with PINTA, is shown in Figure 2a.

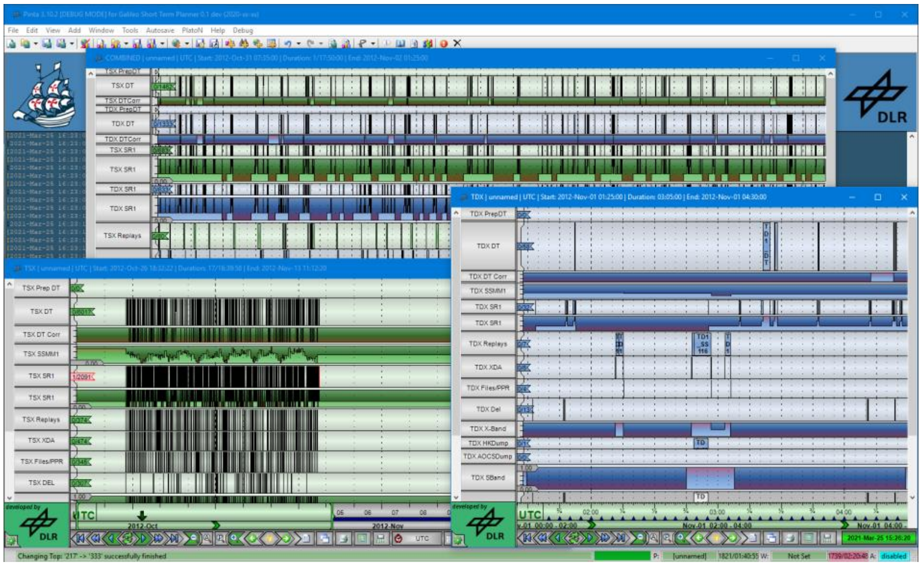

Figure 2a: *The PINTA GUI with three different views, to visualize the TSX/TDX mission.*

The internal PINTA data structure is called the GSOC planning model or simply the "**Project**". In [11], a much more detailed overview of the modelling language can be found. For a better understanding, the most important objects and relations should be briefly outlined:

Each project consists of generic objects such as "**Groups**", "**Tasks**", "**Parameters**" and "**Resources**". These basic objects can in turn be provided with various conditions as well as interdependencies. The central role in every project are the Tasks, which can be scheduled or un-scheduled by adding/removing "**TimelineEntries**" to/from the Project's "**Timeline**". Furthermore, they can be arranged in hierarchical structures with the help of Groups. The most important constrains are "**OrderedTimeDependencies**" between the Tasks, "**UpperBounds**" respectively "**LowerBounds**" of the Resources and "**Allocating-**", "**Accumulating**"- and "**Comparing-ResourceDependencies**" between the Tasks and Resources, with all but the inter-Task constraints being specified via "**Profiles**" over time.
All the dependencies as well as the Resource modifications/comparisons are updated as soon as a Task is scheduled, un-scheduled or modified. In addition, every further task, including its dependencies, as well as the Resources themselves, that are influenced by the change, can indicate conflicts whether or not the modification led to a conflict-free new Project state.

The typical planning model is composed and displayed in PINTA via the so-called "**Project Tree**". As an example, an excerpt of the planning model for the FireBird mission can be seen in **Figure 2b**.

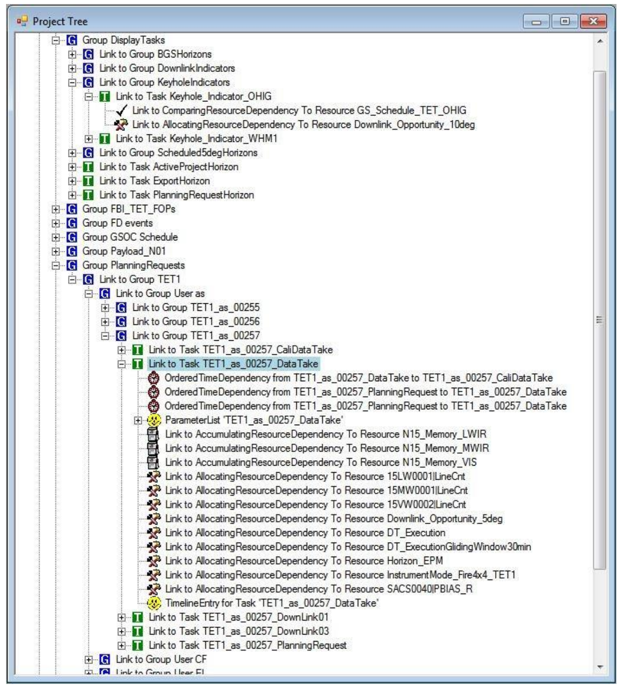

Figure 2b: *The PINTA Project Tree displays an excerpt of the planning model for the FireBird mission.*

In addition to an easy to understand and clear visual representation and the possibility of interactive editing, the possibility to call automatic planning algorithms of the embedded generic planning library PLATO ("**PlanningTool**") or use input of the embedded orbit propagation and event calculation library SCOTA ("**SpaceCraft Orbit and GroundTrack Analysis Tool**") are also included. Furthermore, there is also the option of expanding the scope of functions in a mission-specific manner using a plugin interface.

Figure 2c gives an overview of the whole set-up of PINTA with some of its generic as well as project-specific plug-ins and its relation to our planning library PLATO.

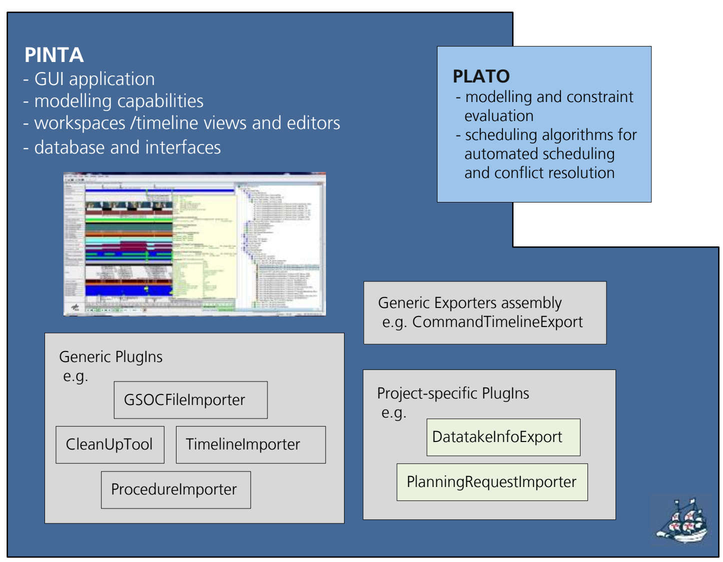

Figure 2c: *The assembly of PINTA*

2.2. PLATO

As already mentioned in the chapter before, PINTA allows for invoking PLATO (“**Planning Tool**”) algorithms. In general, all the fully automated as well as PINTA-based semi-automated mission planning systems at GSOC make use of these scheduling algorithms to automatically schedule and re-schedule indicator tasks, data-takes, downlinks, etc., depending on the mission-specific use case, in order to prepare a conflict-free timeline for the chosen planning horizon.

These scheduling algorithms however are not implemented from-scratch but are composed from the generic algorithm assembly available in the PLATO library. The core idea of this set of algorithms, filters, etc. is to have a reusable, configurable suite of functionalities that can be variously combined and configured and operate on a planning project available in the GSOC modelling language. A more detailed insight into this approach and an overview of most of the available algorithms and filters can be found in [10].

Regarding typical planning problems at GSOC, just some examples for basic algorithms shall be given here and outlined only very shortly to understand the principle:

- **“ChooseValuesToConsider”**: The time range to be forwarded to a sub-algorithm can be determined via various criteria, e.g. respective to the execution time of scheduled horizon Tasks.
- **“ObjectSelection”**: Various filters can be applied to determine to which sub-algorithms which of the currently considered Group(s) and/or Task(s) are to be forwarded in which order.
- **“ValueSelection”**: When having found the next Task to schedule, various filters can be applied to determine whether and with which execution time it is allowed to be scheduled. The invocation of sub-algorithms with time ranges derived from the new TimelineEntry’s execution time is possible.
- **“ConstraintIgnorer”**: This allows temporarily deactivating constraints during the execution of a sub-algorithm, which can be necessary to handle circular dependencies. It enables scheduling and/or un-scheduling Tasks with a potential conflict first before trying to repair the solution by scheduling and/or un-scheduling other Tasks.

Further examples for the application of such PLATO algorithms can be found in more detail in [10].

2.3. SCOTA

The main task of SCOTA (“**SpaceCraft Orbit and GroundTrack Analysis Tool**”) is to provide mission planning with an easy-to-integrate library that computes multiple orbit-related events. Some important examples are the calculation of a satellite's ground coverage ("swath") during a given time interval, the calculation of ground station contact times, or sun-related events such as the satellite moving in and out of the earth's shadow. These calculations require a library with different layers of complexity, including (but not limited to) modules for mathematics, coordinate and time systems, orbit propagation, and geodesics. A primary design goal of SCOTA is to provide this functionality to all projects supported by mission planning as a single generic library.

SCOTA consists of a layered architecture with the following core components:

- SCOTA.Core is the main assembly; it provides the complete SCOTA functionality through a C# API
- The SCOTA.App stand-alone application can be started as a (Windows) executable to process input in XML format
- Two generic SCOTA services encapsulate the access to the *SCOTA.Core* library and thus provide a higher layer of abstraction: SCOTA-Service.Web provides a web service, whereas *SCOTA-Service.ActiveMQ* links SCOTA calculations to an ActiveMQ message broker

2.4. Plug-Ins

The plug-in interface is used to add further generic as well as mission-specific features to supplement the basic PINTA functionality. The most important generic plug-ins, that are frequently used in a typical mission planning system are the following ones:

- The "**CleanUpTool**": Since in many projects, further planning is usually based on the results of previous planning, it is often necessary to clean up before or after each planning run, in order to avoid runtime problems due to excessive storage of information in the underlying database. It can be configured for which model content and at what point in time the cleanup should take place. This ensures that only data that no longer has any influence on the current planning status as well as future planning runs is deleted autonomously, while archiving previous states of the model.
- The "**GSOCFileImporter**": This plug-in is executed to accept input files from other parts of the mission operations ground segment. New Two-Line-Elements and event files from Flight Dynamics, as well as so-called schedule files, are regularly received from the ground station's planning office, and the latest version of each type is processed at the start of every planning run. The content of the event and schedule files is filtered according to the current configuration of the importer and inserted into the planning model as scheduled tasks, including all the constraints which modify the fill level profiles of the corresponding resources.
- The "**ExecutionTimelineImport**": With this plug-in, an already finalized and complete timeline is loaded, which consists of so-called FOPs (Flight Operations Procedures) that were generated in a fully automated planning run, e.g. through the mission planning system used for the TerraSAR-X / TanDEM-X mission [8]. Importing all of this data into PINTA allows for visual cross-checking of the consistency and validity of the timeline, which is supported by a preconfigured set of constraints. Another similar use case was implemented for the FireBird mission, where this functionality was reused to ingest input XMLs from the principle camera investigator, which contain a FOP snippet to be used for so-called SystemOrders. For experimental acquisitions taken for these special requests the camera is then not commanded to switch to one of the standard acquisition modes, but instead it is configured with a FOP containing the ingested snippet.

Furthermore, there are numerous other corresponding plug-ins with very specific functionalities for the various mission planning systems. The only examples to be mentioned at this point are those for the FireBird mission, without going into too much detail: the "**PlanningRequestImporter**", the "**DatatakeInfoFileGenerator**" and the "**EnvelopeRequestImporter**" [7].

Since the plug-in functionality with its dedicated API allows similar additional functions to be easily added, depending on the use cases of the respective project's use cases, the plug-in mechanism has also proven to be very suitable for adding problem-specific functions when it is not part of a mission planning system of a spacecraft mission. Therefore, the plugin interface will play an important role in all following chapters: Mission Planning Systems (chapter III), Sequence of Events Editor (chapter IV), On-Call Tool (chapter V), MuMiCoRoS (chapter VI) and Galileo Planning Tool (chapter VII).

2.5. TimOnWeb

The graphical display tool TimOnWeb ("**Timeline On Web**") is used to visualize the mission timeline, that was previously generated by PINTA - of course with the support of PLATO & SCOTA. It provides users, e.g. the flight directors, insight into the current status of the planning model through a website accessible from the World Wide Web.

Figure 2d shows a snapshot of the current TimOnWeb view for TET, one of the two satellites of the FireBird mission. With its help, all requested, ordered and actually planned activities with all their parameters, as well as ground station contacts, sun and shadow phases, can be displayed in a simple and clear manner, through freely configurable task plots. Further information, such as the utilization of hard disk partitions, battery capacities or even ground station elevation masks and ground tracks, can be plotted as well.

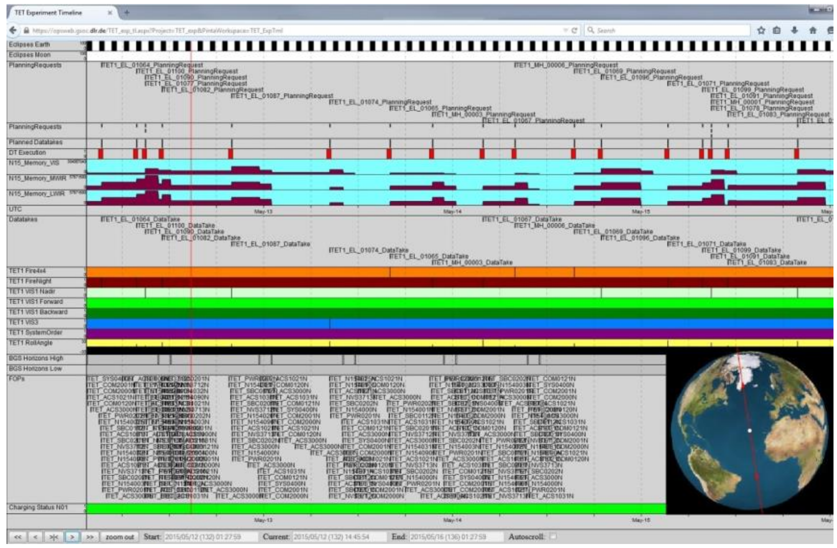

Figure 2d: *Snapshot of TimOnWeb, showing an excerpt of the TET Timeline.*

Unlike PINTA, TimOnWeb does not implement its own planning model representation. The generic server part is based on the PLATO library, so that extended functions of PLATO such as conflict tracking, filter algorithms, etc. can be used. Since TimOnWeb uses JSON files to transfer the data to be displayed from the server to the client, the tool can also be used to display information from other sources and is therefore not exclusively restricted to be used in combination with PINTA.

In addition, the TimOnWeb client uses state-of-the-art web technologies such as HTML5 and WebGL and relies on open source JavaScript libraries, which enables the application to make use of wide-spread and verified generic functionalities such as jQuery, jQueryUI, Moment and satellite-js. The latter is used, for example, to display ground station elevation diagrams and ground track plots, as shown in the lower right corner of **Figure 2d**.

A more detailed description with further information on the technical implementation of TimOnWeb can be found in [3].

III. Mission Planning Systems

The main task of PINTA is to serve as a generic basis for many semi-automatic mission planning systems, for past, current and future spacecraft missions, that are operated at GSOC. **Figure 3a** shows how a typical planning system is structured, how the individual components relate to one another, how everything is embedded within the mission operations ground segment, and which internal as well as external interfaces are commonly used.

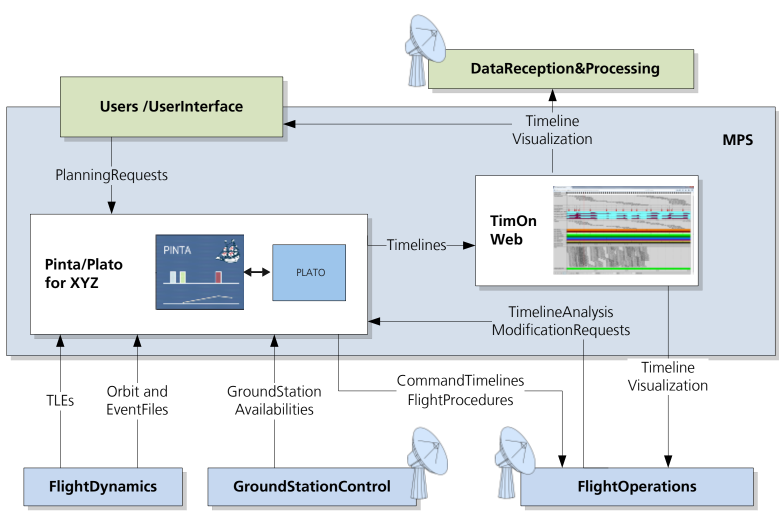

Figure 3a: *Generic setup of a mission planning system with PINTA/PLATO and TimOnWeb.*

PINTA/PLATO in combination with TimOnWeb is used as a semi-automatic and therefore also interactive mission planning framework. Therefore, the daily tasks of an operators engineer typically are: performing and monitoring the planning runs manually as specified in the according ground operations procedure (GOP), execution of recommendations for off-nominal operations, performing additional procedures for special planning tasks for mission specific payloads, and to serve as a user help-desk for inquiries from the customers.

3.1. GRACE

PINTA is currently in its third version. The predecessor of the current generation was PINTA 2, which was used among others during the GRACE mission. This Gravity Recovery and Climate Experiment was a combined mission of DLR and NASA. Two identical satellites performed detailed measurements of earth’s gravity field. The launch was in March 2002 and the end of the science mission was in October 2017. The Main planning Objectives in this mission were the Mass Memory Unit (MMU) Dump to ensure a uninterrupted science data collection, as well as routine activities like sun/moon blinding avoidance and the planning of maintenance and manoeuvre tasks.

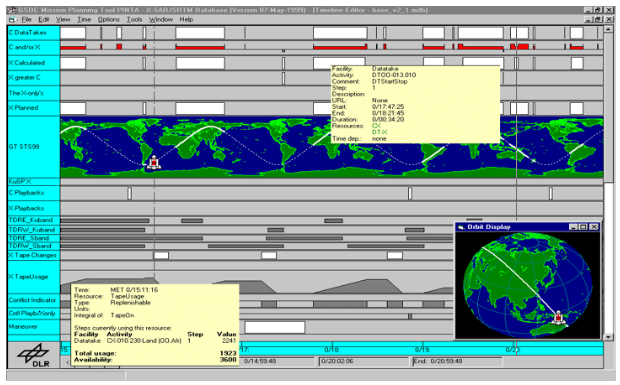

Figure 3b: *Pinta 2 GUI, used during the GRACE mission*

3.2. TerraSAR-X / TanDEM-X

In 2007 the satellite TerraSAR-X (TSX-1) has been launched for the TerraSAR-X mission and in 2010 the mission was completed by the launch of the twin satellite TanDEM-X. When operating in close formation, with distances down to 120m, these radio satellites are capable of creating stereo-scopic SAR-images which can be used to create a digital elevation model.

The corresponding planning problem is one of the most complex, we ever had to deal with, at GSOC. It involves data handling, dump scheduling, instrument sleep-level switching, antenna mode changing, attitude mode switching, etc. All together more than 300 different kind of constraints and over 100 types of resources (>1000 in total) need to be checked for over 10000 activities per day, in order to create a conflict free timeline [8].

One notable difference from all those missions described in this paper is that, in TSTD, PINTA serves as a verification tool only and is not the central planning entity, as the TerraSAR-X/TanDEM-X mission planning system is running unattended as ‘Blackbox’.

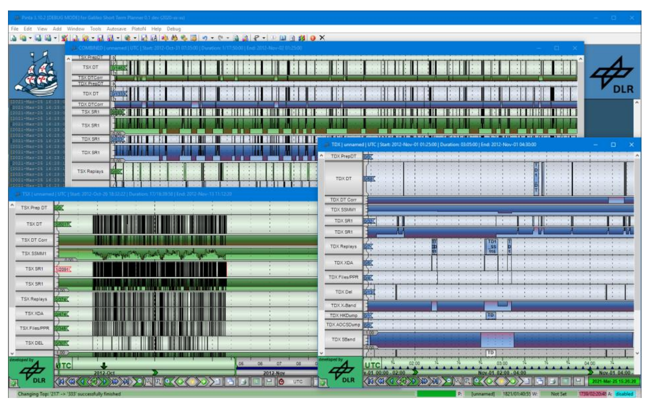

Figure 3c: *PINTA timeline of the TerraSAR-X / TanDEM-X mission*

For the verification of the TSTD mission planning system, the planning products, and in particular the execution timeline with the sequence of Flight Operation procedures (FOPs) and the Saved Stack Files (SSF), is loaded in to PINTA to verify the proper export, and therefore commanding of the space crafts.

3.3. FireBird

The FireBird mission had the goal to deliver science data about worldwide high temperature events to a global science community. It consisted of two infrared, earth observation spacecrafts TET and BIROS. The first one was launched in July 2012 and the second one in June 2016.

The main objective of the mission planning system was the generation of consistent and conflict-free timelines in order to command the payload and background sequence operations. Furthermore, the pre-planning and ordering process for acquisitions of the Infrared Camera System had been supported [6][7].

Figure 3d: *PINTA timeline of the FireBird mission*

3.4. GRACE-FO

The Gravity Recovery and Climate Experiment Follow-On (GRACE-FO) is the successor of the GRACE mission with near-identical hardware, that was launched in May 2018.

Likewise, this mission also consists of two satellites and the main mission planning objective is the prediction of the MMU fill status, the scheduling of the ground stations contacts as well as the handling of the file dumps, for non-science and science data.

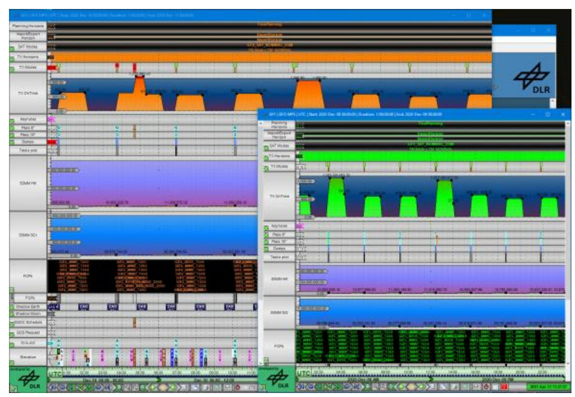

Figure 3e: *PINTA timeline of the GRACE-FO mission*

3.5. EuCROPIS

The mission EuCROPIS (Euglena and Combined Regenerative Organic-food Production in Space) was launched in December 2018. The science mission expired at the end of 2019.

On board the DLR mission were the experiments RAMIS (RAdiation Measurement In Space), the NASA-Experiment Power Cell in Space and an on-board-computer developed by DLR. The mission planning system (see **Figure 3f**) was used to support LEOP-, commissioning- and routine phase with a scheduling of ground station contacts as well as the handling of file dumps for science and non-science data.

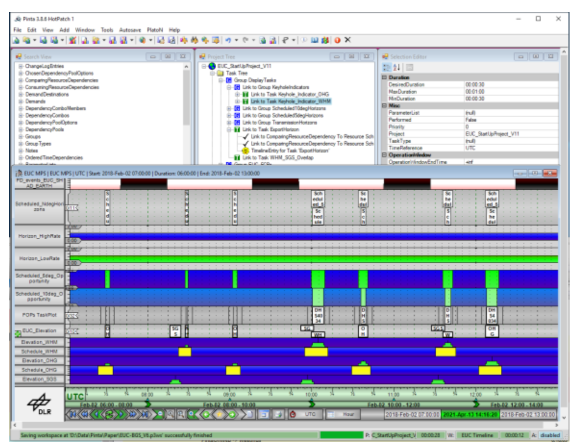

Figure 3f: *PINTA timeline of the EuCROPIS mission*

3.6. PIXL-1 (formerly known as CubeL)

PIXL-1 is the recently launched and first CubeSat where PINTA is used for displaying the timeline and generating the background sequence. The satellite started its mission in January this year and carried a 10x10cm Laser Communication Terminal OSIRIS4CubeSat into the orbit. PIXL-1 is operated by GSOC. The mission planning objectives are the scheduling of ground station contacts, the creation of housekeeping data, GPS data collection and the handling of file dumps. In PINTA all data is transferred and exported in a special developed format (see **Figure 3g**).

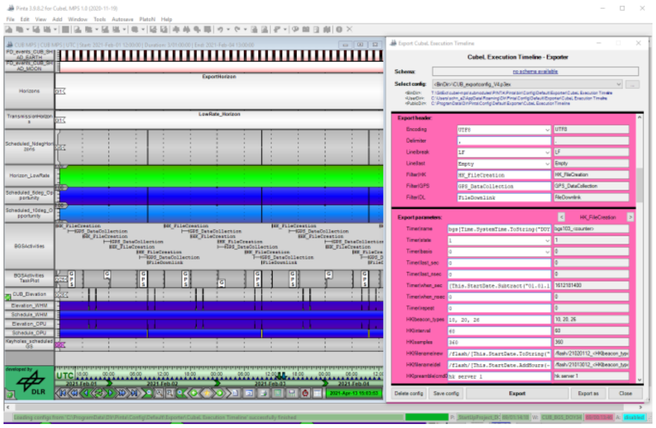

Figure 3g: *PINTA timeline and Execution Timeline Exporter of the PIXL-1/CubeL mission*

IV. Sequence of Events Editor

During the last LEOP's at GSOC, several desired usability improvements for the SoE (Sequence of Events) generation were detected and a lot of new requirements were defined accordingly. It was therefore obvious to consider the possibility of using PINTA for the SoE generation, as well as TimOnWeb for the SoE visualization.

4.1. Overview

In order to meet the special operations requirements during the pre-launch simulation phase, the LEOP phase and the commissioning phase of new satellite missions, many things had to be improved in order to transform PINTA into a usable SoEEditor. One example was the need to extend the PINTA modelling language to handle one main disadvantage with the "OrderedTimeDependencies". Therefore, a template/instance mechanism was introduced for "Tasks" and "Groups" (see [9] for details). Also, an algorithm was implemented to allow the user to reschedule all activities assigned to a conflicting time dependency by only one click or shortcut.

A wide variety of importers had to be developed or modified in order to incorporate FOPs (Flight Operation Procedures), GOPs (Ground Operation Procedures), orbit related events, e.g. ground station visibilities, shadow transitions, or even mission specific events like interfering antenna frequencies with other satellite missions.

In order to reduce the workload for the operator to a minimum, while changing the SoE, it was also necessary to provide an easy-to-use graphical interface. This includes adding FOPs / GOPs and mission-specific events via drag & drop, moving and deleting non-orbit-related events, as well as adjusting time dependencies between different activities.

To share the results of the SoE with LEOP related tools and the operation stuff, several exporters were introduced, reused and extended for e.g. requesting necessary ground station passes or sharing the SoE as PDF files in a graphical and tabular form. An example of the SoEEditor is shown in **Figure 4a.**

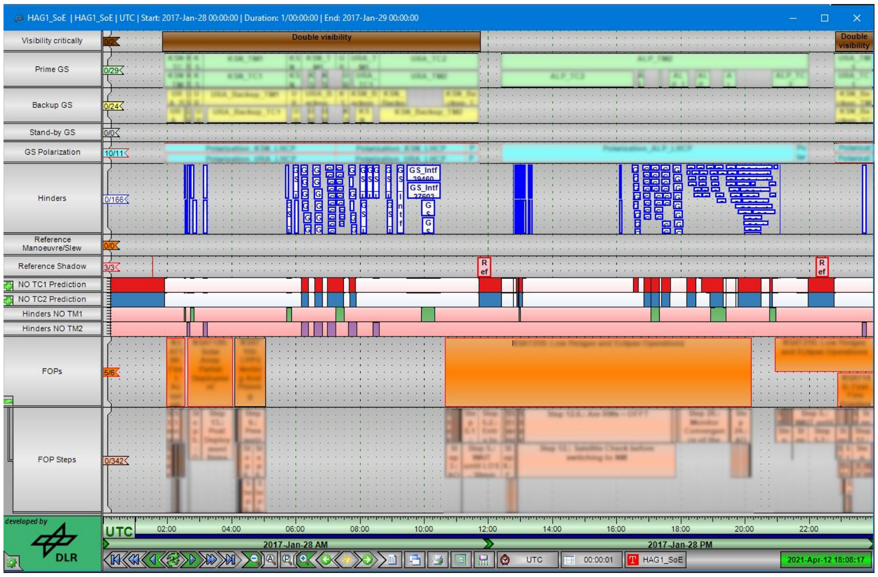

Figure 4a: *Sequence of Events for the LEOP of the GEO mission called HAG. Displayed are the first 24h after the separation of the satellite from the rocket (blurred due to data protection reasons).*

4.2. Displaying

For many missions it is also important that all changes to the SoE are immediately visible in the control room, as well as in the support rooms. For this reason, the interface to TimOnWeb has also been further developed in order to make changes to the timeline immediately visible to all operators (see **Figure 4b**).

Countdown	SimTimeUTC	Duration	Activity	RT	TT	Description	Remark	Shift
FINISHED	2016-119T03:57:00.000Z		LAUNCH					A
0 00:04:57	2016-119T04:16:32.000Z		SEPARATION BIROS					A
0 00:08:25	2016-119T04:20:00.000Z	00:05:00.000	PrePass-Briefing					A
0 00:19:55	2016-119T04:31:29.299Z	00:06:43.400	OHG > 5					A
0 00:21:16	2016-119T04:32:51.000Z	00:00:10.000		BOS_COM3010T_RT		BOS_COM_10_PINGS		A
0 00:22:08	2016-119T04:33:43.000Z	00:00:06.000		BOS_SYS0400N_RT		SYS_NOMINAL_PASS_EHKD_CHECK		A
0 00:23:25	2016-119T04:35:00.000Z	00:00:11.000		BOS_SBC0100L_RT		BOS_SBC_SET_TIME_UTC		A
0 00:24:57	2016-119T04:36:32.000Z	00:00:06.000		BOS_SYS0202N_RT		BOS_SYS_SET_MODE_OPS2		A
0 00:26:38	2016-119T04:38:12.699Z		OHG < 5					A
0 00:43:25	2016-119T04:55:00.000Z	00:05:00.000	PrePass-Briefing					A
0 00:55:52	2016-119T05:07:26.700Z	00:09:24.000	INU > 5					A
0 00:57:25	2016-119T05:09:00.000Z	00:00:07.000		BOS_SYS0220C_RT		SYS_SET_SAFELIST_INDEX	Set to Prime?!	A
0 00:58:53	2016-119T05:10:28.000Z	00:00:37.000		BOS_ACS0361N_RT		BOS_GPS_1_ON		A
0 01:00:26	2016-119T05:12:01.000Z	00:00:02.000		BOS_ACS0353N_RT		BOS_ACS_GPS_LOG_ON		A
0 01:05:16	2016-119T05:16:50.700Z		INU < 5					A
0 01:05:57	2016-119T05:17:32.099Z	00:04:31.000	SGS > 5					A
0 01:10:28	2016-119T05:22:03.099Z		SGS < 5					A
0 01:13:25	2016-119T05:25:00.000Z	00:12:00.000	PostPass-Briefing					A
0 02:18:25	2016-119T06:30:00.000Z	00:05:00.000	PrePass-Briefing					A
0 02:30:42	2016-119T06:42:16.300Z	00:08:15.700	INU > 5					A

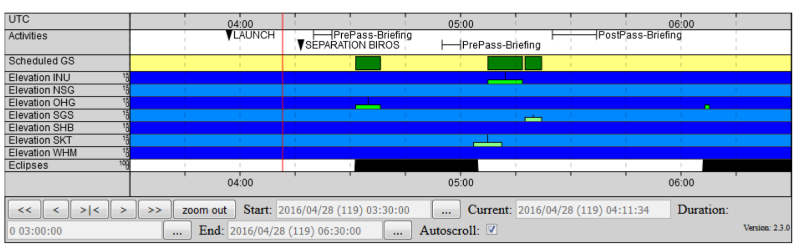

Figure 4b: *Sequence of Events for the LEOP of a LEO spacecraft called BIROS. The upper part shows an alphanumeric table with all the past, currently active and upcoming events, during the first ground station contacts after the separation of the spacecraft. The lower part is a graphical presentation of the events on the mission timeline.*

A more detailed overview of the interaction between Pinta and TimOnWeb with regard to the SoE generation during LEOPs is given to the interested reader in [3].

V. On-Call Tool

PINTA’s original purpose was the planning of satellite mission timelines. However, the generation of those timelines is not the only planning problem encountered in regular satellite operation. One such problem is the on-call shift scheduling for multiple subsystems of one or more satellites. The difficulty here lies in the various dependencies and constraints of people and subsystems like person outages, role-conflicts and so on. The on-call tool is used to generate conflict-free role-based shift plans.

5.1. Overview

The on-call plan at the GSOC is usually automatically planned for one year in advance but with the option to re-plan every quarter. For this purpose, PINTA with a specialized PLATO algorithm is used. This algorithm is configured to create a plan where every subsystem role of every satellite which requires on-call support has one person planed at all times. It also has to keep the constraints such as outages of persons, role-conflicts, legal and company requirements in mind and not violate those. There are also optional goals for the algorithm to improve the plan for the staff members. One for example is that the on-call shifts should last a whole week to prevent daily changes of the person who is on-call. Such optional goals are nice to have but not always achievable.

After the automatic planning run, the on-call plan can be displayed, analyzed and manually modified. Such manual modifications of an existing plan are easily done with PINTA using the TimelineView (example plan see **Figure 5a**). The plan will then be published and displayed on TimOnWeb so that the on-call staff gets the information and the operator knows who to call in case of a contingency.

The operational plan can also be modified by the on-call-coordinator using the timeline in PINTA at any time. These modifications are usually requested by the on-call staff themselves. A change may be necessary if a person on shift becomes unavailable. This ensures that the plan is always up to date.

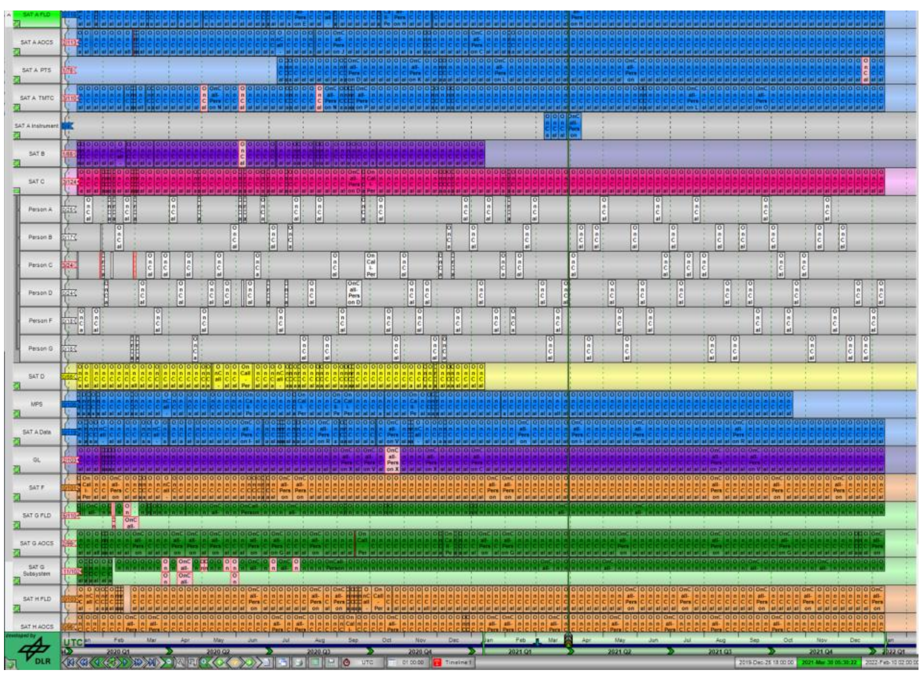

Figure 5a: *Example of an On-Call plan over the period of two years with several different satellites and subsystems.*

5.2. Plugins

There are two special plugins written for the On-Call Tool. The first one help prepare the on-call project for planning. It provides the user with a GUI to add, delete or change staff members and stations. In this context, station stands for a subsystem of a satellite that requires on-call service. This on-call plugin relieves the on-call coordinator by adding, removing and modifying of resources, tasks and dependencies required for the planning and maintaining of the plan. The second plugin is embedded in the Task-Editor GUI and is opened by simply double-clicking of any on-call task within the Timeline View of PINTA. It can be used to easily (de-)select different subsystems of an on-call task of a person.

VI. MuMiCoRoS

With an increasing number of satellite missions, a point will eventually be reached where it is no longer possible to treat all missions separately. A new tool was needed to coordinate the ground station planning of the various missions at GSOC, to avoid resource conflicts, such as antenna utilization and operator availability.

6.1. Overview

The "**Multi-Mission-Control-Room-and-Pass-Scheduler**" (MuMiCoRoS), based on PINTA, was developed for the weekly ground station planning of all LEO missions at GSOC: TerraSar-X, TanDEM-X, TET, Biros, Grace-FollowOn 1 & 2 and Eu:CROPIS.

It quickly became clear that PINTA was the right software for this task, since it was already possible out-of-the-box to import all required files, such as flight dynamics events, or antenna availability, as these are already part of every mission planning system. The possibility of exporting request files, which in turn is required for ground station booking, was also already available, because this was a necessary requirement for the SoEEditor. In addition, the GUI is flexible enough to be able to be adapted to this planning problem (see **Figure 6a**).

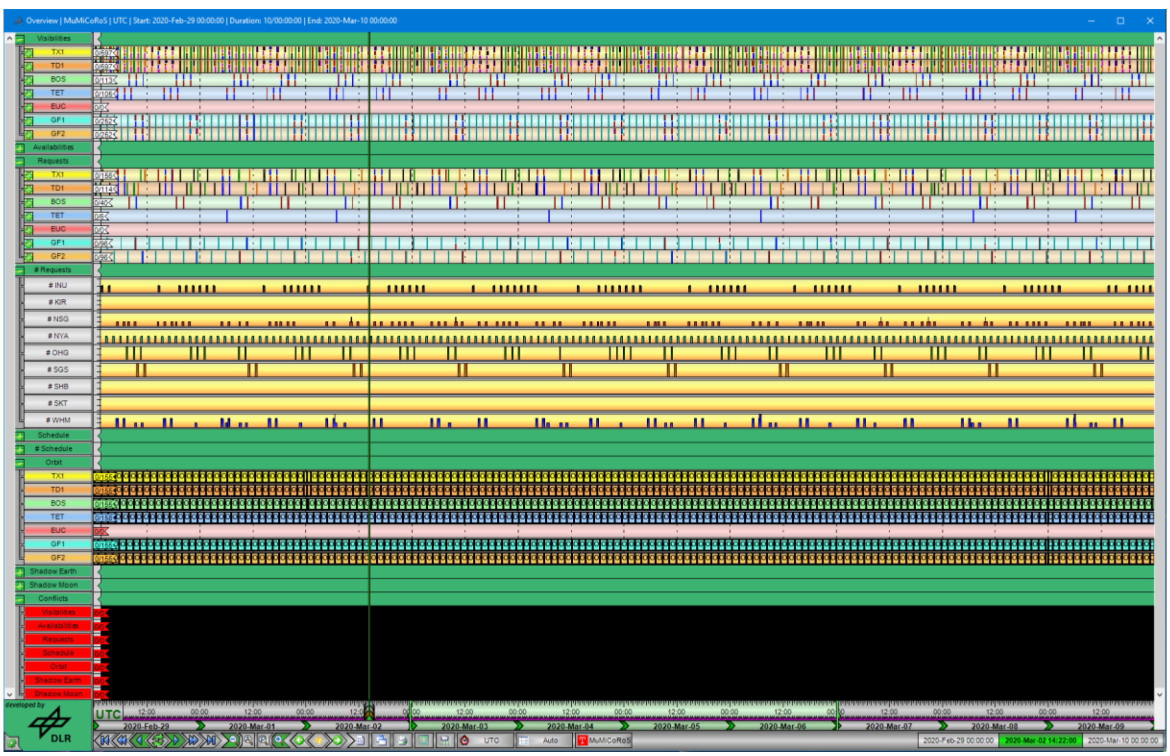

Figure 6a: *The main timeline view of MuMiCoRoS. It shows all visibilities for all satellites and ground stations, the availabilities of the ground stations and the currently requested contacts. Resource conflicts, which are very rare and not existent in this plan, are indicated in the bottom plots.*

6.2. Plugins

Although PINTA already provided a wide range of functions that were necessary for MuMiCoRoS, an additional plugin was still required to add further functionality. The so-called "**GS-Scheduling**" plugin includes a planning algorithm that is able to generate a conflict-free plan for all missions in a very short amount of time. Each mission has an individual planning strategy, which often changes over time due to changed mission objectives. In addition, it must be guaranteed that the missions do not get in each other's way.

The automatic planning does its job very well and rarely requires manual intervention. Yet, due to unforeseeable events, like broken antennas, to name just the most obvious one, it is still necessary to have the possibility to perform the planning process manually. For this reason, during the development of MuMiCoRoS, great value was placed on the fact that a user can change plans very comfortably and have an overview of all possible side effects. The GS-Scheduling plugins embeds a ground station scheduling specific panel into the Task-Editor's GUI (see **Figure 6b**). By simply double-clicking on any task within the Timeline View of PINTA, this panel appears and can be used to unscheduled, schedule or edit a property of the requested contacts. Another tool that makes planning easier for the user is the GS Scheduling Editor (see **Figure 6c**), which can be used for manual scheduling or for getting another look at the plan in tabular form, with the possibility to filter subsets of the whole plan.

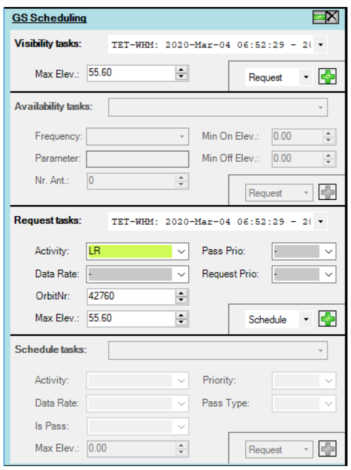

Figure 6b: *GS Scheduling plugin embedded within the Task Editor.*

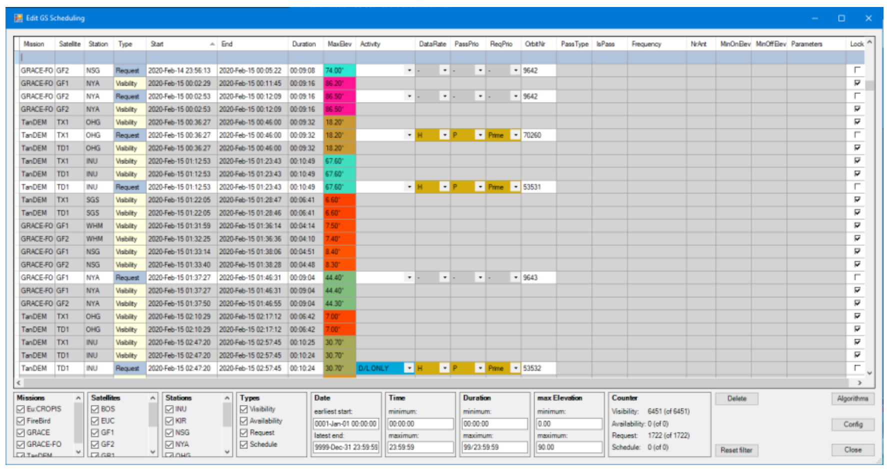

Figure 6c: *GS Scheduling plugin for manual scheduling, triggering an automatic scheduling run, or simply for having a better clarity over the whole timeline in tabular form with the possibility to apply filters.*

VII. Galileo Planning Tool

Galileo is one of the four Global Navigation Satellite Systems (GNSS) worldwide, which went live in 2016. The (not yet) fully established constellation and infrastructure of Galileo will consist of 30 satellites distributed over three orbital planes, and about 20 sensor stations spread across the globe for monitoring the navigation signals. Furthermore, two Galileo Control Centers (GCC-D & GCC-I), one located in Germany (Oberpfaffenhofen) and one located in Italy (Fucino), the GNSS service center in Spain (Torrejón de Ardoz) and several mission uplink stations and S-Band Stations, are required for satellite monitoring and control.

The Galileo program is under the responsibility of the European Commission, which has delegated responsibility for the provision of the Galileo service to the GNSS Service Agency (GSA). The service provision comprises various contracts, including the Galileo Service Operations (GSOp) contract, which was awarded to Spaceopal GmbH in 2016. The Munich based, Spaceopal GmbH is a joint venture founded in 2009 by the two partners DLR Gesellschaft für Raumfahrtanwendungen (GfR) mbH and Telespazio S.p.A., which itself is a joint venture owned by Leonardo and Thales Group. As a service operator of the Galileo system, Spaceopal GmbH offers users worldwide high-quality navigation and time measurement. Its planning team works with multiple subcontractors to ensure service operations for the Galileo fleet of spacecrafts and the infrastructure. The primary goal herby is of course, to ensure that operation, maintenance and further development of the system is performed smoothly and without interruptions at any time, so that a continuous global navigation service can be provided. Users all around the world are already using Galileo for single and dual frequency positioning down to the m-level [1].

PINTA, among various competitors, was the tool of choice by the Spaceopal Planning Team which fits their needs the closest. It is used to support the Galileo Service Operation (GSOp) since 2018 and will also be referred to as the Galileo Planning Tool (GPT) in this context of this chapter. The following subchapters will first give an overview of the associated planning problem and then describe technical implementation aspects in more detail. To get a more comprehensive overview of the Galileo planning process see [1] and [2].

7.1. Overview

The planning process at Galileo includes three timelines that differ in granularity (see **Figure 7a**). These timelines are divided into four plans that are constantly updated within firmly defined moving timeframes:

- **"LTP"**: A Long-Term Plan for the next 15 months that contains all the high-level activities with a rough allocation of time.
- **"MTP"**: Two Mid-Term Plans (MTP-OPE & MTP-VAL) for the next 15 weeks, one for the operational chain (OPE) and one for the validation chain (VAL).
- **"STP"**: A Short-Term Plan for the next ten days with the best granularity and therefore highest temporal accuracy down to minutes.

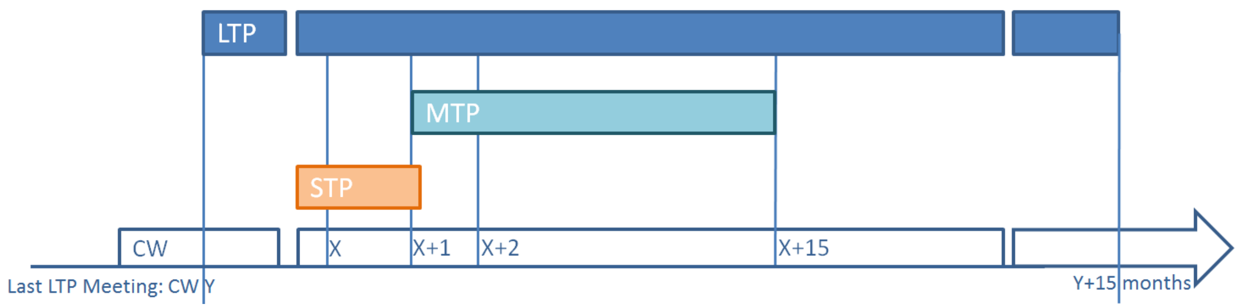

Figure 7a: *Coverage of the LTP, MTP and STP timeframes, which is shifted every calendar week.*

Activities that are scheduled during these timelines cover all Galileo subsystems: flight ops, control segment, mission segment, remote locations, service operations, hardware, software, hosting, network, etc…

For obvious reasons, these plans are interdependent and it must be ensured that all information details are shared correctly between all the different plans at all times. The most reliable way, to achieve this goal, is that these plans must be processed using the same tool, which also provides functionalities that can be used to link the plans together. Furthermore, displaying activities in a clear and easy to grasp way, are a basic requirement to reduce human error. This requires a hierarchical and logical arrangement of the activities, to provide a quick overview about impacts, relationships, delays and especially planning states like approvals, cancellations and the final execution of the activities. An example of the main Timelines within the GPT can be seen in **Figure 7c** and one of the color-coding definitions is shown in **Figure 7b**. Similar ones exist for LTP and STP. A better insight into how much effort was required to bring all these plans together is explained in [1].

Figure 7b: *GPT states color coding schema for the MTP.*

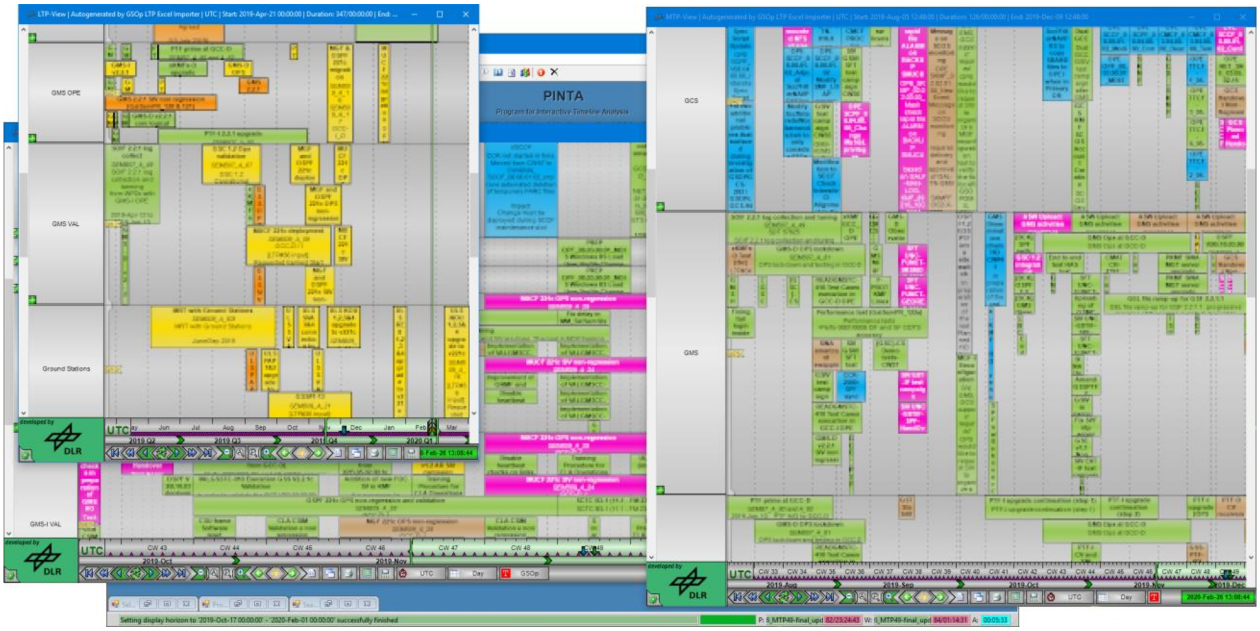

Figure 7c: *Screenshot of the GPT showing the LTP (left), MTP-VAL (middle) and MTP-OPE (right) in different views and the main window in the background (task boxes were pixelated due to data protection reasons).*

Although PINTA already provided a large number of generic functionalities due to its development history, especially as a mission planning system, it was still necessary to undertake further generic as well as specific extensions to support the Galileo service operations. These will be dealt within the following.

7.2. Generic Exporter

Due to the continuously growing number of projects PINTA has to handle and the hereby involved requirements to support more and more different export formats, the decision was made to completely redesign its export functionalities. This redesign has already been described in [3] together with the interaction between the exporter and the PINTA scripting language, which in turn provides the functionality that every single export value can easily be configured within the GUI. All possible values from the whole PINTA data model structure or even any arithmetic or logic combination of these values can be defined as export values via this scripting language.

In order to comply with the requirements for being a usable Galileo Planning Tool, the generic exporter was further improved. With the primary goal to generate graphical exports, with the same look and feel as the appearance of PINTA timeline views (see **Figure 7c**), the following file formats were implemented:

- **"PINTA 4 Workspace":** Creates the database files (*.p4ws) for TimOnWeb workspaces (version < 2.6), based on the configuration of a pre-defined PINTA timeline view.
- **"TimOnWeb2 Workspace":** Generates a corresponding database file (*.t2ws) for newer TimOnWeb workspaces (version >= 2.6) with an even better match in appearance.
- **"Graphical Pdf":** Creates a PDF-Document (*.pdf), again with the intention to mimic the appearance of a PINTA timeline view.

These exporters together with the already existing “**XLSX Task Table**” exporter provide all necessary file formats, which are required to distribute the current schedules of the LTP and the two MTP chains to hundreds of mail recipients after each planning meeting, in three different file formats.

To look uniform and clear, a lot of effort has been put into harmonizing the exported pdf-files (see **Figure 7e**), the xlsx-files (see **Figure 7d**) and the configuration of the TimOnWeb workspace (see Figure **Figure 7f**) with the GPT workspace (see Figure **Figure 7c**).

MTP Output for CW09
#

Start CW	Start Date	End CW	End Date	Resource	Activity Name	SDT Ticket	Description	Status
CW01	30.12.2019	CW12	19.03.2020	GSMC/FR				To Be Re-Planned
CW08	16.02.2020	CW17	19.04.2020	Ground Stations/GSS/FUC_GSS KIR_GSS				Approved for Planning
~~CW08~~	~~17.02.2020~~	~~CW11~~	~~08.03.2020~~	~~GSAT / 0220~~				~~Canceled~~
CW09	23.02.2020	CW11	08.03.2020	GMS/GMS_D & GSMC/FR				Approved for Planning
CW09	24.02.2020	CW10	02.03.2020	GSAT / 0216				Approved for Planning
CW10	02.03.2020	CW10	02.03.2020	Ground Stations/ULS/KOU_ULS				Approved for Planning
CW10	02.03.2020	CW10	04.03.2020	GSAT / 0204				Approved for Planning
CW10	02.03.2020	CW10	04.03.2020	GCS/GCS_D / FDF_D				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0205 0206 0210				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GMS/GMS_I/GACF_I MGF_I MUCF_I				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0102 0103				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0201 0202 0214 0216 0218 0222				Approved for Planning
CW10	02.03.2020	CW11	08.03.2020	GSAT / 0207 0212				Approved for Planning
CW10	02.03.2020	CW14	29.03.2020	Remarks				Pending Approval
CW10	02.03.2020	CW19	03.05.2020	Remarks				Approved for Planning
CW10	03.03.2020	CW10	03.03.2020	Ground Stations/ULS/KOU_ULS				Approved for Planning
CW10	03.03.2020	CW10	03.03.2020	GMS/GMS_D/TSP_D				Approved for Planning
CW10	03.03.2020	CW10	05.03.2020	Ground Stations/ULS/PAP_ULS				Approved for Planning
CW10	03.03.2020	CW10	06.03.2020	GMS/GMS_D/OSPF_D				Approved for Planning
CW10	04.03.2020	CW10	04.03.2020	Ground Stations/ULS/KOU_ULS				Approved for Planning
CW10	05.03.2020	CW10	05.03.2020	Ground Stations/TTCF/NOU_TTCF				Open

Figure 7d: *Screenshot of the XLSX Tasks Exporter output (blanked fields and removed columns due to data protection reasons).*

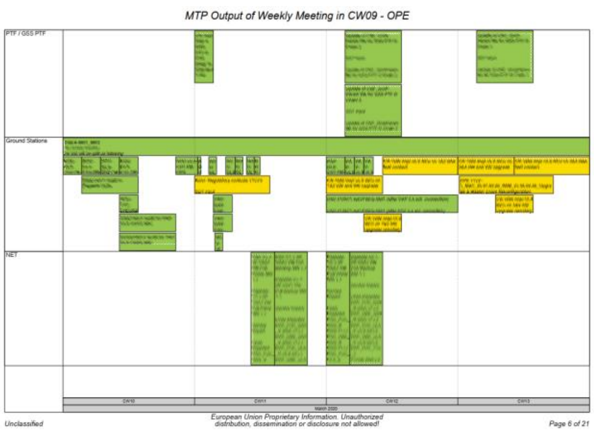

Figure 7e: *Screenshot of an example page of the PDF-Exporter output (blurred due to data protection reasons).*

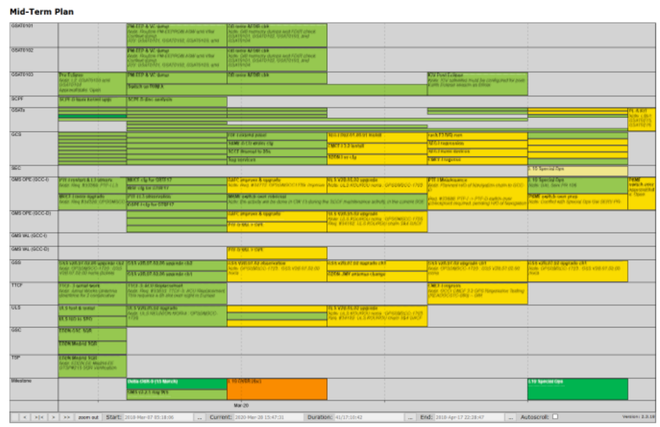

Figure 7f: *Screenshot of the the TimOnWeb workspace with MTP data provided by the GPT (blurred due to data protection reasons).*

7.3. Change-Log Exporter/Importer

When transferring data from one database into another one by hand it can be very time consuming and is also a source of human error with high probability, especially when losing concentration. This kind of manual interaction needs to be avoided in any case. For that reason, PINTA has a built-in exporter and importer for transferring database changes from one PINTA project to another one.

The “**Change-Logs”** are the foundation of the “**Undo/Redo”** concept. Every single change and accordingly the complete history of changes within the project are tracked separately and saved as so-called “**Change-Log-Entries”**. Whenever a project is saved, the Change-Log-Entries are saved as well and every change can be reversed even after restarting the program.

Every modification is represented as one specific change log of one of the following types:
- Changed: An object property has been changed.
- Added: A new object has been added.
- Deleted: An object has been deleted.

The built-in “**Change-Log-Editor”** provides functionalities to visualize**,** filter**,** undo**,** redo**,** enable**,** disable and clean-up Change-Logs for the current project (see **Figure 7g**). The “**Change-Log-Exporter”** can be used to export change logs that were made within the project. These files can then be used as patches, to update the database of another project.
Accordingly, the “**Change-Log-Importer”** is used to import these changes and merge them into the database of the current project.

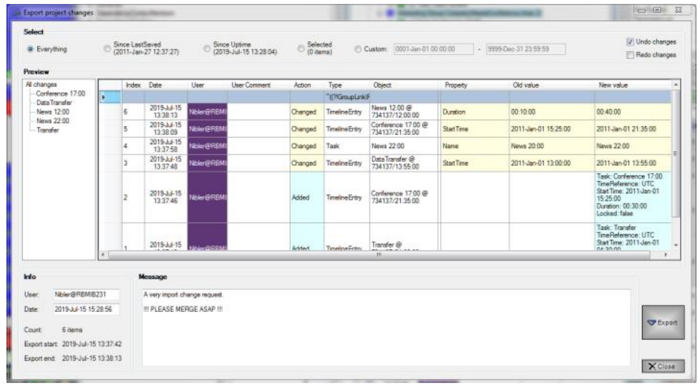

Figure 7g: *Screenshot of the PINTA Change-Log exporter, showing a patch with changes that can be exported.*

Merging change-sets, means synchronising database, which is a tough problem in computer science. If for example multiple patches are imported which are assumed to be created by different people, there is always a high chance for conflicts, probably because nobody is aware of the changes made by everybody else. This could lead to the same property being modified in two different incompatible ways, to an object getting deleted twice, or to an object getting changed, which has already been deleted by another user. To deal with this kind of problems a user interaction is required to fix the problem in one or another way.

- **Ignore**: The current conflict will be ignored by disabling the Change-Log-Entry. The state of the database will not be affected by this change.
- **Repair**: If changes lead to a situation were different valid states are possible, the user can interactively decide which solution to consider as the correct one.
- **Cancel**: The merging of the current change will be cancelled and the whole patch will be reverted, without causing any changes to the database.

Specifically, for the GPT, the functionality of the Change-Log Exporter and Importer had to be expanded even more. Although the export and import of patches works smoothly, the problem hereby is, that the content of the files is not in a human-readable form, but very technical. A way had to be found to make patches readable, and to make things even more complicated, even changeable by the user. For that reason, an interface was developed to the so-called “**MTP-Tool**”, an EXCEL macro sheet, developed by Spaceopal. A detailed description of this tool and corresponding workflows can be found in [1].

7.4. Service-Tree Importer

The “**Service-Tree**” is a hierarchical tree-structure for all ground, navigation and space related services that are available at Galileo. The granularity of this tree can be as deep as seven or more hierarchy levels, resulting in hundreds of entries. Every activity within the GPT can be assigned to any number of services. To make things even more challenging, the structure of this tree is not fixed, but changes on a regular basis.

Fortunately, this flexible tree-like behavior is one of the basic principles of how the PINTA database is designed. Briefly described, the model consists of objects like “**Groups**”, “**Tasks**”, “**Parameters**” and “**Resources**” which are connected with “**Constraints**”. All these model objects can be arranged in a hierarchical group structure, as can be seen in **Figure 7h**. For a more detailed overview of the GSOC planning modelling language have a look at [3] or [11].

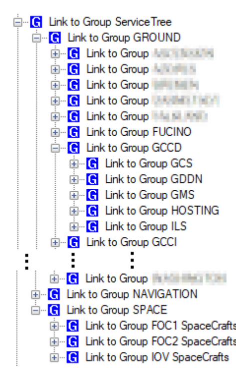

Figure 7h: *PINTA Project Tree, containing the whole Galileo Service Tree.*

In order to keep up with the changes to the Galileo Service Tree, it was therefore necessary to develop a plugin that automatically applies updates to the PINTA Project Tree.

7.5. Task-Editor

Within PINTA the Task-Editor is the main tool to display the details of individual tasks and to make manual modifications. Most of the already available generic functionalities like changing the name or comment, writing notes, moving the tasks start and end time, or tracking changes of individual users are very helpful but were not enough to support the Galileo planning process.

As already mentioned in the subchapter about PINTA-Plugins (see **Figure 7i**) it is possible to embed mission-specific extension panels into the Task-Editor’s GUI.

The so-called “**GSOp PR-Editor**” is a new plugin that was written to fulfil the requirements for the GPT. Not only simple extensions such as various ticket IDs, additional description fields, information about the originator, or flags like “Service Impact”, “Urgency” or “Hazardous Condition”, with pre-defined values had to be implemented. To make the user's work easier, faster, and less prone to errors, more complex functionality was needed.

The “**Service Tree**” section contains the full tree-structure of all the services that are available at Galileo. And every one of them can be assigned or unassigned to every activity, which also affects all child services in a recursive way. This can conveniently be done by selecting and clicking a button.

Within the “**Family Tree**” all the parent-child relationships of activities are shown, even if they belong to different timelines, and highlighted according to their planning state. Furthermore, this tree can also be used to create or delete these relations between activities easily via drag & drop.

The “**States**” section can be used to change the planning state of the current activity. Besides, it also shows a summary of all the planning states of all parent and child activities, which gives the user a fast and easy to grasp overview of the relationships.

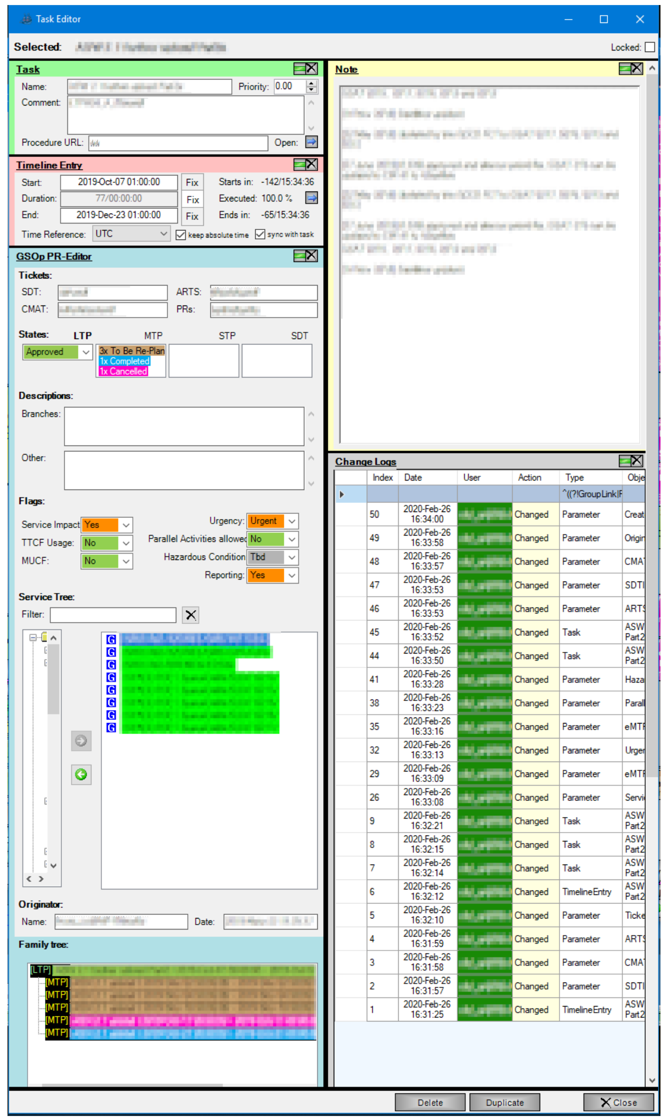

Figure 7i: *Screenshot of the the GPT Task-Editor, with the embedded PR-Editor plugin (blurred due to data protection reasons).*

7.6. SDT Interface

Most of the inputs for the GPT, mainly for MTP-VAL and MTP-OPE, come from another tool, the so-called Service Desk Tool (SDT). Basically, all the activities, like support, resource bookings, system changes, maintenance, etc., that take place at Galileo are tracked within SDT.

As a logical consequence, to make manual data exchange superfluous, an interface to the SDT is the next step, which is currently under development. This would complete the entire flow of information from the creation of a ticket, the approval process, the execution and the final reporting to an overall system. This topic is explained in more detail in [2] and [1].

From the implementation point of view for the GPT this means that in addition to LTP, MTP and STP an additional timeline for SDT is required. Since this hardly differs from the existing plans in terms of modeling, graphical representation and required user interaction, the effort for this is relatively low. However, the export or import of the data is more complicated in order to keep the GPT and the SDT in a permanently consistent state. The work on the interface will therefore continue to be challenging…

VIII. Conclusions and Outlook

In summary, we can say that we see ourselves as well positioned with our GSOC mission planning tool suite that has been developed within and around PINTA. During decades of mission planning experience at GSOC, PINTA has proven to be flexible, reliable and generically applicable for solving various mission planning problems with a different degree of automation. The range of applications here is not restricted to serving single spacecraft missions, but also comprises solving multi-mission planning problems like the allocation of on-call shift personnel and supporting the sharing of ground resources such as control room and ground station availabilities, or the Galileo service operations planning, as has been explained in this paper.

Furthermore, PINTA can be easily combined with the other components and libraries of the GSOC mission planning tool suite, such as the timeline web viewer TimOnWeb, the planning library PLATO, the event-calculation library SCOTA, or the so-called "Swath Preview and Ordering Tool" (SPOT) [7]. Of course, that is just as important as the well-established integration with the other GSOC sub-systems, such as flight dynamics, satellite and ground station monitoring and control, via generic interfaces.

What is not to be underestimated is the fact that the approach of having a comfortable in-house development for all the above - mentioned mission planning tool suite is much appreciated. This has proven to be beneficial as it easily allows for including direct feedback from operations personnel as well as other mission responsible persons and creates expertise for estimating probable future requirements. And what should also not be kept secret, even if nobody likes to talk about it, is that the biggest advantage of controlling your own software is that debugging and troubleshooting is much easier und way faster.

While in the last years the focus of the ongoing evolution of PINTA lay on modernization, enhancing the robustness and extending the features, also the enhancement of the user friendliness became more and more important. This comprises simplifying workflows, improving the GUI functionalities, performance enhancements, provision of simplified and extended information export capabilities, and, last but not least, documentation. For instance, e.g. a convenient handbook and the provision of trainings and instruction material have to be promoted.

Among the big milestones for the next years, one is to establish the connectivity to the upcoming generic GSOC "**Reactive Planning framework**" (formerly known and presented as the "**Incremental Planning System**" [4]), making PINTA an optional front-end of this, to allow for having interactive mission planning systems also basing on the same integrated, widely configurable and scalable tool suite as already foreseen for the fully automated use cases. Further, not less important, development tasks will be accompanying common developments of the GSOC modelling language as well as the international standardization of interfaces and mission planning and scheduling services in the CCSDS environment [5].

Nevertheless, in parallel, of course, there are always modifications for specific project and user needs wherever requested and viable. Having the control about the development of the software tool-suite it is ensured that further modifications, feature enhancements and functionality extensions will not be missed to stay well-prepared for and in the future as well. Above all, one big future goal is the integration of PINTA and TimOnWeb into one application becoming "**PintaOnWeb**".

Lots of ambitious waters have been sailed so far [12], but the journey won't end here. Not too far away and already visible on the horizon is the next adventure, when PINTA will surf the web via PintaOnWeb.

To be continued …

References

[1] S. Brogl, S. Manailescu, J. Gutierrez, R. Ballweg, F. Mrowka; "How Galileo Planning became automated"; *16th International Conference on Space Operations, SpaceOps 2020; Cape Town; South Africa; 18th – 22nd May 2020; SpaceOps-2020,6,2,1x249;* 2020.

[2] V. Carandente, S. Brogl, R. Cadenas Gorgojo, R. Roßgotterer, D. Docal Lareu, R. Ballweg; "Integrated GSOp Planning Architecture"; *16th International Conference on Space Operations, SpaceOps 2020; Cape Town; South Africa; 18th – 22nd May 2020; SpaceOps-2020,6,2,2x246;* 2020.

[3] R. Nibler, F. Mrowka, M. T. Wörle, J. H. Hartung, C. Lenzen; "PINTA and TimOnWeb – (more than) generic user interfaces for various planning problems"; *10th International Workshop on Planning and Scheduling for Space;* 2017.

[4] M.T. Wörle, C. Lenzen, T. Göttfert, A. Spörl, B. Grishechkin, F. Mrowka and M. Wickler, "The Incremental Planning System - GSOC's Next Generation Mission Planning Framework," *13th International Conference on Space Operations;* 2014.

[5] "Mission Planning and Scheduling Concept, Draft Green Book," *Report Concerning Space Data System Standards;* 2016.

[6] A. Spörl, C. Lenzen, M.T. Wörle, J.H. Hartung, A. Braun and M. Wickler, "Mission Planning System for the TET-1 OnOrbitVerification Mission," *13th International Conference on Space Operations;* 2014.

[7] M.T. Wörle, A. Spörl, J. H. Hartung, C. Lenzen and F. Mrowka, "The Mission Planning System for the Firebird Spacecraft Constellation," *14th International Conference on Space Operations;* 2016.

[8] C. Lenzen, M.T. Wörle, F. Mrowka, M.P. Geyer and R. Klaehn, "Automated Scheduling for TerraSAR-X/TanDEM-X," *7th International Workshop for Planning and Scheduling in Space;* 2011.

[9] J.H. Hartung, R. Nibler, C. Peat, A. Spörl, M.T. Wörle and C. Lenzen, "GSOC SoE-Editor 2.0 - A Generic Sequence of Events Tool," *14th International Conference on Space Operations;* 2016.

[10] C. Lenzen, M.T. Wörle, F. Mrowka, A. Spörl and R. Klaehn, "The Algorithm Assembly Set of Plato," *12th International Conference on Space Operations;* 2012.

[11] "GSOC Planning Modelling Language," *http://www.dlr.de/rb/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-MIB/GSOC_Modelling_Language.pdf,* cited on 2017-06-07.

[12] Wikipedia article "Pinta (ship)," *https://en.wikipedia.org/wiki/Pinta_(ship).*

[13] C. Stangl, B. Lotko, M.P. Geyer, M. Oswald and A. Braun, "GECCOS – the new Monitoring and Control System at DLR-GSOC for Space Operations, based on SCOS-2000"; 2014.

[14] J.P. Hamacher, T. Beck "ProToS: Automation of Flight Control Procedures for the European Data Relay System"; 2018.

[15] M.P. Geyer, F. Mrowka, C. Lenzen "TerraSAR-X/TanDEM-X Mission Planning - Handling Satellites in Close Formation"; 2010.